

PROC TEMPLATE and its Application in Tables & Crosstabulation Tables

Ruchi Prakash Sahu, Syneos Health, Hyderabad, India

ABSTRACT

PROC TEMPLATE enables us to customize the appearance of SAS ODS outputs. We can create, extend, or modify existing templates for various types of output for different attributes like: styles, tables, crosstabulation tables, columns, headers, footers, tagsets and ODS Graphics.

As we all know all tabular output generated by SAS is created using a table template. This Table template describes how columns, headers and footers appear in the table. Since tables are generated in SAS using table templates, we can use TEMPLATE Procedure and edit the table attributes to modify the output produced.

This paper will discuss the Basics of the PROC TEMPLATE, its syntax and limitations. Further it will discuss in detail (using examples) how we can change the appearance of the table output generated using PROC UNIVARIATE. Additionally, we will edit headers within the table template and sets column attributes to enhance the appearance of the HTML output. Most of us have used the PROC TEMPLATE for creating the ODS Graphics, this paper focus more on "Crosstabulation Table Templates", "Style Template" and "Tabular Templates".

INTRODUCTION

Although many people don't realize it, almost all tabular output generated by SAS is created using a table template. Table templates describe what columns, headers, and footers should appear in the table. Additionally they can contain style overrides to specify extra style information. There are ways to compute new columns based on columns in the existing data set. SAS generates tables using table templates, we can modify the output of procedures by editing the table templates that the procedures use. This paper is based on exploring different functionality of PROC TEMPLATE by giving some examples and application of its various features.

Syntax

```
PROC TEMPLATE;
  DEFINE COLUMN <...>END;
  DEFINE FOOTER <...>END;
  DEFINE HEADER <...>END;
  DEFINE STYLE <...>END;
  DEFINE TABLE <...>END;
  DEFINE TAGSET <...>;
  DEFINE EVENT <...>END;
  DEFINE CROSSTABS <...>END;
  DEFINE STATGRAPH <...>END;
  DELETE template-path <...>;
  EDIT template-path<...>END;
  TEST DATA=data-set </
  STORE=libref.tmplt-store>;
RUN;
```

STYLE

Describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style determines the overall appearance of the documents that use it. Each style consists of style elements.

Style elements are collection of style attributes that apply to a particular part of the output.

For example, BACKGROUND= attribute specifies the color for the background of an HTML table, and the FONTSTYLE= attribute specifies whether to use a Roman, a slant, or an italic font.

Syntax

```
DEFINE STYLE style-path |
Base.Template.Style </
STORE=libref.template-store>;
  PARENT=style-path;
  NOTES "text";
  CLASS style-element-name(s)<"text">
</ style-attribute-specification(s)>;
IMPORT style-specification <media-
type-1 <, media-type-2> ...>;
STYLE style-element-name(s) <FROM
style-element-name | _SELF_ > <"text">
</ style-attribute-specification(s)>;

END;
```

DEFAULT STYLE FOR HTML OUTPUT

Default Style for HTML output: ODS uses styles to display the procedure or DATA step results. We can modify the layout of the output by customizing these styles. The first HTML output is generated from PROC PRINT using the default style.

Second output is produced with the same HTML output from PROC PRINT with a small change the style option. Fig 1.1 is using default style for HTML output i.e styles.HTMLBlue, Fig 1.2 is using "Journal2" for HTML destination type.

Similarly for destination type Listing and RTF outputs default style is Listing and RTF respectively.

PhUSE US Connect 2018

```
/*Default Style*/
ods listing close;
ods html file="style.html";
TITLE "List of Adverse Events";
```

Fig1.1

List of Adverse Events	
AE Reported Term	System Organ Class
Diarrhea	Gastrointestinal disorders
Gastrointestinal Infection	Infections and infestations
Infection	Infections and infestations
Nausea	Gastrointestinal disorders
Respiratory Infection	Infections and infestations
Urinary Tract Infection	Infections and infestations
Vomiting	Gastrointestinal disorders

```
/*Default Style*/
ods listing close;
ods html file="style.html" style=Journal2;
TITLE "List of Adverse Events";
```

Fig1.2

List of Adverse Events	
AE Reported Term	System Organ Class
Diarrhea	Gastrointestinal disorders
Gastrointestinal Infection	Infections and infestations
Infection	Infections and infestations
Nausea	Gastrointestinal disorders
Respiratory Infection	Infections and infestations
Urinary Tract Infection	Infections and infestations
Vomiting	Gastrointestinal disorders

EXAMPLE#2: CHANGING THE STYLE ATTRIBUTES

Using the default style, we get the output as Fig1.1 with following code:

```
ods listing close;
ods html file="style.html";
TITLE "List of Adverse Events";
proc sql;
    select UNIQUE(AETERM),AEBODSYS FROM DATA.ADAE;
QUIT;
ODS LISTING;
```

In this example, alterations in the "cellcontents", "header", "systemtitle" and "table" statements are used to customize the HTML output.

```
/*Example 2: Custom Style */
proc template;
define style style1;

    style cellcontents / fontfamily="helvetica" fontweight=medium
        backgroundcolor=blue fontstyle=roman fontsize=3 color=grey;

    style header / backgroundcolor=#BFCFFF fontfamily="arial"
        fontweight=medium fontstyle=roman fontsize=3 color=white;

    style systemtitle / fontfamily="arial"
        fontweight=bold backgroundcolor=white
        fontstyle=roman fontsize=4 color=salmon;

    style table /borderspacing=5
        borderwidth=7;
end;
```

```
ods listing close;
ods html file="style.html" style=style1;
TITLE "List of Adverse Events";
proc sql;
    select UNIQUE(AETERM),AEBODSYS FROM
        DATA.ADAE1;
QUIT;
ODS LISTING;
```

Fig 1.3

List of Adverse Events

AE Reported Term	System Organ Class
Diarrhea	Gastrointestinal disorders
Gastrointestinal Infection	Infections and infestations
Infection	Infections and infestations
Nausea	Gastrointestinal disorders
Respiratory Infection	Infections and infestations
Urinary Tract Infection	Infections and infestations
Vomiting	Gastrointestinal disorders

EXAMPLE#3: USE OF INHERITANCE & CELLSTYLE

Using the existing style change in specific attributes can be attained with the inheritance(PARENT=), it will copy all of the style elements from the parent style to the new child style.

CELLSTYLE Specify which style element and style attributes to use for different values in the column

```

/*Change in style using inheritance*/
define style style2;
    parent = style1;

    style cellcontents from cellcontents /
    fontweight=medium
    backgroundcolor=blue
    fontstyle=roman
    fontsize=4;

    style footer from header/
    fontsize=2
    TEXTALIGN=LEFT;
end;
run;

proc template;
    define table table_cellstyl; mvar sysdate9;
        dynamic colhd; classlevels=on;
    cellstyle _val_ IN ("Gastrointestinal Infection" "Respiratory Infection")
        as {background=blue color=white};
end;
run;

ods html close;
ods html file="style.html" style=style2;
TITLE "List of Adverse Events";
FOOTNOTE "*Row in BLUE are Adverse Events of Special Interest";
proc sql noprint;
create table ae as
select UNIQUE(AETERM),AEBODSYS FROM DATA.ADAE1;
QUIT;

data _null_;
    set ae;
    file print ods=(template="table_cellstyl");
    put _ods_;
run;

ods html close;
ods html;

```

TABLES

The TEMPLATE procedure enables us to customize the tabular appearance of SAS output. Tabular templates describe how tables should be constructed. This includes the content and placement of headers and footers, the content and placement of columns, and style overrides. All SAS procedures, except PROC PRINT, PROC REPORT, and PROC TABULATE, use tabular templates to describe how their tabular output should look. This means that you can change the structure of tables that are generated by SAS procedures by using tabular templates. With the TEMPLATE procedure, we can create and modify tabular templates.

Tabular templates include the following types:

- column templates
- header templates

Fig 1.4

List of Adverse Events

AE Reported Term	System Organ Class
Diarrhea	Gastrointestinal disorders
Gastrointestinal Infection	Infections and infestations
Infection	Infections and infestations
Nausea	Gastrointestinal disorders
Respiratory Infection	Infections and infestations
Urinary Tract Infection	Infections and infestations
Vomiting	Gastrointestinal disorders

*Row in BLUE are Adverse Events of Speical Interest

Inheritance can be initiated by the PARENT= statement or the FROM= option in the STYLE statement.

"file print ods=(" statement routes the DATA step results to ODS and use the Table_cellstyl table template.

Likewise EDIT statement enables us to edits an existing template.

PhUSE US Connect 2018

- footer templates
- table templates

TABLES EXAMPLE 1

The default output of the frequency table is presented in fig 2.1 with missing values for AEBODSYS and AETERM.

Fig 2.1

Use of Translate statement and Numeric cell value categorization

Obs	TRT01A	AEBODSYS	AETERM	COUNT
1	Treatment A			2
2	Treatment A	Gastrointestinal disorders	Diarrhea	1
3	Treatment A	Gastrointestinal disorders	Nausea	1
4	Treatment A	Infections and infestations	Gastrointestinal Infection	1
5	Treatment A	Infections and infestations	Respiratory Infection	1
6	Treatment A	Infections and infestations	Urinary Tract Infection	1
7	Treatment B			3
8	Treatment B	Gastrointestinal disorders	Diarrhea	2
9	Treatment B	Gastrointestinal disorders	Nausea	1
10	Treatment B	Gastrointestinal disorders	Vomiting	3
11	Treatment B	Infections and infestations	Gastrointestinal Infection	2
12	Treatment B	Infections and infestations	Infection	1
13	Treatment B	Infections and infestations	Respiratory Infection	5
14	Treatment B	Infections and infestations	Urinary Tract Infection	1

Fig 2.2

Use of Translate statement and Numeric cell value categorization

Treatment Group	System Organ Class	AE Reported Term	Frequency Count
Treatment A	Uncoded AE term	Uncoded AE term	2
	Gastrointestinal disorders	Diarrhea	1
		Nausea	1
	Infections and infestations	Gastrointestinal Infection	1
		Respiratory Infection	1
Treatment B		Urinary Tract Infection	1
	Uncoded AE term	Uncoded AE term	3
	Gastrointestinal disorders	Diarrhea	2
		Nausea	1
		Vomiting	3
	Infections and infestations	Gastrointestinal Infection	2
		Infection	1
		Respiratory Infection	5
		Urinary Tract Infection	1

The following example shows the result of using a TRANSLATE statement in table template that alters the specific value/content of a cell value attributes, color attributes of the text value and change in color based on the frequency values in the cell.

```
proc template;
    define table table_style;
        translate _val_="" into "Uncoded AE term";
        notes "ADAE with Uncoded AETERM";

    define column char_var;
        generic=on;
        blank_dups=on;
        cellstyle _val_ in ("Gastrointestinal Infection" "Respiratory Infection")
            as {color=red fontweight=bold};
    end;

    define column num_var;
        generic=on;
        justify=on;

        cellstyle _val_ <= 1 as data {color=green fontstyle=italic},
            1 < _val_ <= 3 as data
            {color=orange fontstyle=italic},
            3 < _val_ <= 10 as data
            {color=red fontweight=bold};
        end;
    end;
run;

proc freq data=adae_ucae noprint;
    tables trt01a*aebodsys*aeterm / missprint out=freq_tbl(drop=PERCENT);
run;
```

generic = specifies that multiple variables can use the same column template.

justify= Align the values in the column without regard to the format field.

Translate = Specify that missing values show the text "Uncoded AE term" in the report

cellstyle _val_ : Specify which style element and style attributes to use for different values in the column.

End; & run; : END statement ends the table template. The RUN statement ends the PROC TEMPLATE step.

CROSSTABULATIONS

Using TEMPLATE we can control and change the appearance of crosstabulation (contingency) tables that are created with the PROC FREQ.

PhUSE US Connect 2018

By default, crosstabulation tables are formatted according to the CrossTabFreqs template that SAS provides. However, you can create a customized CrossTabFreqs table template by using the `TEMPLATE` procedure with the different statements in the for attributes shown in the below fig 3.1.

We can create a custom crosstabulation and can do following:

- specify a style for each value in a cell
- change the stacking order of values in a cell
- change and style headers and footers
- use custom formats for cellvalues
- use variable labels in headers and footers
- style table regions independently
- change or remove the legend

Syntax

```
PROC TEMPLATE;
    DEFINE CROSSTABS table-path </
STORE=libref.template-store>;
    <table-attribute-1; <table-attribute-n>;
```

Regions that can be modified:

Fig 3.1

AE CrossTab										
The FREQ Procedure										
Frequency Percent Row Pct Col Pct	Table of TRT01A by AETERM								Header/ STYLE	
	Row variable Name/ ROWS_HEADER	AETERM(AE Reported Term)						Column variable name/ COLS_HEADER		
	TRT01A(Treatment Group)	Diarrhea	Gastrointestinal Infection	Infection	Nausea	Respiratory Infection	Urinary Tract Infection	Vomiting	Total	
	Legend/ LEGEND_STYLE	Treatment A	1	1	0	2	2	1	0	7
			4.00	4.00	0.00	8.00	Data cell/ CELL_STYLE 8.00	4.00	0.00	28.00
14.29			14.29	0.00	28.57	28.57	14.29	0.00		
33.33			25.00	0.00	50.00	28.57	50.00	0.00		
Treatment B	2	3	2	2	5	1	3	18		
	8.00	12.00	8.00	8.00	20.00	4.00	12.00	72.00		
	11.11	16.67	11.11	11.11	27.78	5.56	16.67			
	66.67	75.00	100.00	50.00	71.43	50.00	100.00			
Total	3	4	2	4	Column total/ COL_TOTAL_STYLE 7	2	3	25	Grand total/ GRAND_TOTAL_STYLE	
	12.00	16.00	8.00	16.00	28.00	8.00	12.00	100.00		

CROSSTABULATIONS

The syntax used to create crosstabulation templates differs significantly from other table templates. The crosstabulation template has no parent template, and it cannot serve as a parent to another template. Crosstabulation table templates use `DEFINE CELLVALUE` blocks instead of the `DEFINE COLUMN` blocks that are used in other table templates. Likewise `CELLVALUE` statement is used instead of `COLUMN` statement in other table templates.

EXAMPLE

Fig 3.2 : Default output

Frequency of AE by Treatment										
Frequency Percent Row Pct Col Pct	Table of TRT01A by AETERM									
	TRT01A(Treatment Group)	AETERM(AE Reported Term)								
		Diarrhea	Gastrointestinal Infection	Infection	Nausea	Respiratory Infection	Urinary Tract Infection	Vomiting	Total	
		Treatment A	1	1	0	2	2	1	0	7
			4.00	4.00	0.00	8.00	8.00	4.00	0.00	28.00
			14.29	14.29	0.00	28.57	28.57	14.29	0.00	
		33.33	25.00	0.00	50.00	28.57	50.00	0.00		
Treatment B	2	3	2	2	5	1	3	18		
	8.00	12.00	8.00	8.00	20.00	4.00	12.00	72.00		
	11.11	16.67	11.11	11.11	27.78	5.56	16.67			
	66.67	75.00	100.00	50.00	71.43	50.00	100.00			
Total	3	4	2	4	7	2	3	25		
	12.00	16.00	8.00	16.00	28.00	8.00	12.00	100.00		

Fig 3.3

Frequency of AE by Treatment

Table of AE with Frequency Category								
Treatment Group	Diarrhea	Gastrointestinal Infection	Infection	Nausea	Respiratory Infection	Urinary Tract Infection	Vomiting	Total
Treatment A	1	1	0	2	2	1	0	7
	4.00	4.00	0.00	8.00	8.00	4.00	0.00	28.00
	14.29	14.29	0.00	28.57	28.57	14.29	0.00	
	33.33	25.00	0.00	50.00	28.57	50.00	0.00	
Treatment B	2	3	2	2	5	1	3	18
	8.00	12.00	8.00	8.00	20.00	4.00	12.00	72.00
	11.11	16.67	11.11	11.11	27.78	5.56	16.67	
	66.67	75.00	100.00	50.00	71.43	50.00	100.00	
Total	3	4	2	4	7	2	3	25
	12.00	16.00	8.00	16.00	28.00	8.00	12.00	100.00

Frequency Category of AE

Green - % 0- 10

Orange- % 10- 20

Red - % > 20

```

/*Example : Crosstabs*/
ods path (prepend) work.templat(update);
ods noproctitle;

proc template;
  define style crosstb1;
    parent=styles.htmlblue; /*Edit attributes over htmlblue style*/
    style body /
      backgroundcolor=white;
    style systemtitle /
      backgroundcolor=white
      fontsize=4
      fontweight=bold
      fontstyle=italic;
    style systemfooter /
      backgroundcolor=white
      fontsize=2
      fontstyle=italic;
    style proctitle /
      backgroundcolor=white
      color=#6078bf
      fontweight=bold
      fontstyle=italic;
  end;

  define crosstabs Base.Freq.CrossTabFreqs;
    notes "Crosstabulation table";

    style=table {backgroundcolor= white};
    cell_style=data {backgroundcolor=white};
    row_total_style=data {backgroundcolor=salmon};
    col_total_style=data {backgroundcolor=salmon};
    grand_total_style=datastrong
      {backgroundcolor=salmon};
    rows_header=RowsHeader cols_header=ColsHeader;
  end;

```

The **DEFINE** statement creates the crosstabulation template Base.Freq.CrossTabFreqs in the work library

The define Crosstabs statement attributes modify the appearance of individual table regions. Each attribute corresponds to a specific region of the table. Ex. Style, cell style etc.

ROWS_HEADER= style attribute specifies RowsHeader as the header for rows.

COLS_HEADER= style attribute specifies ColsHeader as the header for columns.

PhUSE US Connect 2018

```
label = "Frequency Counts and Percentages";
define header TableOf;
  text "Table of AE with Frequency Category";
  style=header { backgroundcolor=#BFCFFF color=#6078bf fontstyle=italic};
end;

define header RowsHeader;
  text _ROW_LABEL_ ;
  style=header {backgroundcolor=#BFCFFF color=#6078bf fontstyle=italic};
end;

define cellvalue Frequency;
  header="";
  label="Frequency Count";
  format=BEST7.; data_format_override=on; print=on;
end;

define cellvalue Expected;
  header="";
  label="Expected Frequency";
  format=BEST6. data_format_override=on print=on;
end;

define cellvalue Deviation;
  header="";
  label="Deviation from Expected Frequency";
  format=BEST6. data_format_override=on print=on;
end;

define cellvalue TotalPercent;
  header="";
  label="Percent of Total Frequency";
  format=6.2 print=on;
end;

define cellvalue Percent;
  header="";
  label="Percent of Two-Way Table Frequency";
  format=6.2 print=on;
end;

define cellvalue RowPercent;
  header="";
  label="Percent of Row Frequency";
  format=6.2 print=on;

  cellstyle _val_ <= 10 as datastrong {color=green},
    10 < _val_ <= 20 as datastrong {color=orange},
    _val_ > 20 as datastrong {color=red};

end;

define cellvalue ColPercent;
  header="";
  label="Percent of Column Frequency";
  format=6.2 print=on;
end;

define cellvalue CumColPercent;
  header="";
  label="Cumulative Percent of Column Frequency";
  format=6.2 print=on;
```

The **DEFINE CELLVALUE** statements define the values that will appear in the cells of the crosstabulation table. The **HEADER=** attribute specifies the text that appears in the legend.

The **CELLSTYLE AS** statement uses expressions to set the style element of the cells conditionally according to the values of the variables for the Frequency cellvalue. The **_VAL_** variable represents the value of a cell.

The **CELLVALUE** statement specifies which cellvalues appear in the output.

Specify which headers and footers appear in the output. The **HEADER** statement specifies which header templates are applied to your output. The **FOOTER** statement specifies which footer templates are applied to your output.

PhUSE US Connect 2018

```
        end;  
        cellvalue  
        Frequency  
        TotalPercent Percent  
        RowPercent ColPercent CumColPercent;  
        header TableOf ControllingFor;  
        footer NoObs Missing;  
    end;  
run;  
  
ods html file='AECtabs.html' style=crosstb1;  
title "Frequency of AE by Treatment";  
..... <footnotes statements> ...  
proc freq data=data.adae;  
    tables trt01a*aeterm / missprint;  
run;  
ods html close;
```

DELETE STATEMENT:

Once we have created the output using custom template, custom templates/styles can be deleted if no longer needed.

```
proc template;  
    delete <template-name>;  
run;
```

Ex:

```
proc template;  
    delete Base.Freq.CrossTabFreqs;  
run;
```

CONCLUSION

PROC TEMPLATE can help us control and improve the appearance of the output in many ways. There are many other different statements and options available which are not included here in this paper.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank my managers Neelam Shinde, Namrata Murthy and Supriya Deshmukh for encouraging me to write the paper and providing the guidance and support at regular intervals of time. Last but not the least I would like to acknowledge Gulshan Sonboir who has done a remarkable work in providing the critical view on the topic contents and examples.

RECOMMENDED READING

SAS® 9.4 Output Delivery System: Procedures Guide, Third Edition

DISCLAIMER

The content of this paper is the works of the author and do not necessarily represent the opinions, recommendations, or practices of Syneos Health.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name	: Ruchi Prakash Sahu
Company	: Syneos Health
Address	: Unit 301A&B, Building 12B, Raheja Mind Space, Madhapur
City / Postcode	: Hyderabad, Telangana 500081
Work Phone	: +91 40 47571974
Email	: ruchi.sahu@syneoshealth.com
Web	: syneoshealth.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Brand and product names are trademarks of their respective companies.